



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
[www.uspto.gov](http://www.uspto.gov)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/741,986	12/20/2000	David A. Eatough	10559/376001/P10182	8003
20985	7590	11/01/2005	EXAMINER	
FISH & RICHARDSON, PC 12390 EL CAMINO REAL SAN DIEGO, CA 92130-2081				YIGDALL, MICHAEL J
ART UNIT		PAPER NUMBER		
2192				

DATE MAILED: 11/01/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

<b>Office Action Summary</b>	Application No.	Applicant(s)
	09/741,986	EATOUGH ET AL.
	Examiner Michael J. Yigdall	Art Unit 2192

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

#### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

#### Status

- 1) Responsive to communication(s) filed on 04 August 2005.
- 2a) This action is FINAL.                  2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

#### Disposition of Claims

- 4) Claim(s) 1-28 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) Claim(s) \_\_\_\_\_ is/are allowed.
- 6) Claim(s) 1-28 is/are rejected.
- 7) Claim(s) \_\_\_\_\_ is/are objected to.
- 8) Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

#### Application Papers

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on \_\_\_\_\_ is/are: a) accepted or b) objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

#### Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) All    b) Some \* c) None of:  
 1. Certified copies of the priority documents have been received.  
 2. Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.  
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

#### Attachment(s)

- |  |   |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892)   | 4) <input type="checkbox"/> Interview Summary (PTO-413)                     |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)                                   | Paper No(s)/Mail Date. _____  |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)<br>Paper No(s)/Mail Date _____ | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
|  | 6) <input type="checkbox"/> Other: _____                                    |

## **DETAILED ACTION**

1. This Office action is in response to Applicant's submission filed on August 4, 2005.

Claims 1-28 are pending.

### *Response to Arguments*

2. Applicant's arguments have been fully considered but they are not persuasive.

At the outset, it is noted that one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981), and *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986). Here, the rejections are based on a combination of Foster and Davis.

Applicant contends that the statement presented in the previous Office action misconstrues the claimed subject matter and fails to address each and every limitation of the claim (Applicant's remarks, page 3, first full paragraph).

However, the examiner does not agree with Applicant's characterizations. The plain language of the claims does not exclude the teachings of Foster and Davis. Specifically, in Davis, it is the configuration file that determines the list of installation tasks (see, for example, FIG. 3). In other words, it is the configuration file that "provides" the list of installation tasks. For example, the configuration file provides the answer to step 81, which asks, "Is component in typical install?" If so, the component is included in the list of installation tasks (step 82). Furthermore, a list of installation tasks that is input to a task manager is considered a script. In such a case, the list of installation tasks "scripts" the operation of the task manager.

Regardless, Foster expressly discloses creating and distributing a software package (see, for example, 3, lines 47-52), as presented in the previous Office action. The software package includes a control file (see, for example, package 200 in FIG. 2), and the control file provides a script to install, upgrade and remove the software package (see, for example, column 8, lines 39-55). This software package, created in preparation for later distribution and installation (see, for example, column 12, lines 25-35), is considered an “X-package” or an “X-package document.” The element that Foster does not expressly disclose is an intermediate “package importer” or an intermediate step of “importing” a package from a vendor to create the software package based on the vendor’s control file. Davis teaches this element in terms of “importing” a configuration file from a third party (see, for example, column 3, lines 44-51 and column 4, lines 1-9). The configuration file of Davis is analogous to the control file of Foster, and is considered a “vendor package template.”

Applicant contends that the Office action has provided no evidence that Davis describes a package importer that creates an X-package document based on a vendor package template, or that Davis does so in preparation for later software distribution and installation (Applicant’s remarks, page 3, last paragraph).

However, Foster creates a software package or an X-package document in preparation for later software distribution and installation, as noted above. The package includes a control file that provides a script. Davis, in turn, suggests a package importer to create the package based on an analogous configuration file or vendor package template that provides an analogous “script” of installation tasks.

Applicant contends that the cited portions of Davis describe an actual installation of a software package on a target computer, and not importation of a software package, such as importation into a distribution management server that can occur before selection of any particular installation to a target computer (Applicant's remarks, page 3, last paragraph to page 4, top paragraph).

However, Foster discloses a distribution management server (see, for example, column 12, lines 13-24), as presented in the previous Office action. The software packages that Foster creates for later distribution and installation are distributed from this server. In fact, the packages are created before any package is selected for installation on a target computer (see, for example, column 12, lines 25-35).

Furthermore, the examiner recognizes that FIG. 3 of Davis illustrates an installation process. Nonetheless, to reiterate the previous Office action, the actual installation is not performed until step 94. Several other steps are completed before step 94, such as opening the configuration file (step 71), getting parameters from the configuration file (step 76), and processing the configuration file in a loop (step 77 to step 85). Although these steps are part of an installation process, separately they are an "importation" process that is performed before any components are installed at step 94.

Applicant contends that Davis is clearly describing software that runs on a target computer and manages the installation process on that target computer, and that this cannot be equated with "a package importer to receive said at least one software package, where said package importer creates an X-package document based on said vendor package template in

preparation for later software distribution and installation” (Applicant’s remarks, page 5, first full paragraph).

However, as noted above, Foster discloses creating an X-package document in preparation for later software distribution and installation. Davis suggests incorporating a package importer to do so based on a vendor package template.

Applicant contends that Davis clearly fails to disclose the claimed format of the X-package document created in preparation for later software distribution and installation (Applicant’s remarks, page 6, last paragraph to page 7, top paragraph).

However, the recited “format” is merely one “that makes said X-package manageable in a software package management system independent of vendor-specific aspects of the at least one software package” (claim 1). Indeed, the “importation” process of Davis enables this feature. Davis expressly discloses that the system “provides a common interface for use for the installation of third party software and would allow the users to customize their configuration of third party applications being installed” (see, for example, column 1, lines 54-57), as Applicant emphasizes (Applicant’s remarks, page 6). This common interface is a reason to incorporate the “importation” process into Foster. Certainly the “format” of a software package created in such a manner permits, and does not preclude, the common interface. Moreover, Foster discloses a format that includes a common set of attributes (see, for example, column 7, lines 35-45).

It is noted that although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

Applicant contends that Foster's description of including in a control file of a software package the name, address and other contact information for the person responsible for maintaining or creating the software package (i.e., the identity of the package creator) cannot be equated with an importing user (Applicant's remarks, page 7, second full paragraph).

However, the examiner disagrees. For example, Foster discloses a user who creates the software package, as Applicant notes. Davis suggests that to create the software package, the user "imports" a package. Thus, the identity of the user who creates the software package is considered the identity of the "importing user" as well.

***Claim Rejections - 35 USC § 103***

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 1-26 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 6,675,382 to Foster (art of record, "Foster") in view of U.S. Pat. No. 6,279,154 to Davis (art of record, "Davis").

With respect to claim 1 (previously presented), Foster discloses a software management system (see, for example, the abstract), comprising:

(a) a package agent to receive, deploy and execute an X-package at a target computer (see, for example, FIG. 4 and column 9, lines 19-46, which shows the installation, i.e. the

Art Unit: 2192

deployment and execution, of a software package, and column 12, lines 25-35, which shows that the software package may be received from a remote server).

Although Foster discloses creating and distributing a software package (see, for example, column 3, lines 47-52), as well as the means to install, upgrade and remove a package based on a control file (see, for example, column 7, lines 46-51) that provides a script comprising a list of commands (see, for example, column 8, lines 39-49), Foster does not expressly disclose a vendor package template that provides a script to install, upgrade and remove the package, and a package importer to create an X-package document based on the vendor package template in preparation for the distribution and installation of the software.

However, Davis discloses a third-party component configuration file, i.e. a vendor package template (see, for example, column 3, lines 44-51), and the means to import the configuration file or template to create a corresponding list of installation tasks, i.e. a script, in preparation for later installation (see, for example, column 4, lines 1-9 and 20-24, and column 5, lines 28-32). The packages are manageable in a system that enables users to install and configure third-party applications using a single, common format or interface (see, for example, column 1, lines 43-57), independent of the installation process that is specific to the third party or vendor (see, for example, column 1, lines 22-34).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the vendor package template and package importing features taught by Davis with the system of Foster, in order to provide a common interface for managing software packages from a plurality of third parties or vendors.

Foster in view of Davis further discloses the limitation wherein the script comprises a list of commands in a programming language (see, for example, column 12, lines 56-67, which shows using a computer programming language such as Perl so that the same script can execute on many platforms without recompilation).

With respect to claim 2 (previously presented), Foster in view of Davis further discloses the limitation wherein said package importer receives an importing user identity for recording (see, for example, column 8, lines 31-33, which shows receiving the identity of the package creator, i.e. the importing user).

With respect to claim 3 (original), although Foster discloses vendor-supplied software packaging systems (see, for example, column 2, lines 8-13), Foster does not expressly disclose the limitation wherein said at least one software package includes packages from different vendors.

However, Davis discloses an installation system that includes packages from a plurality of third parties or vendors and enables users to manage them using a single, common interface (see, for example, column 1, lines 43-57).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to include packages from different vendors in the Foster system, in order to provide a common interface for managing software packages from the plurality of sources as taught by Davis.

With respect to claim 4 (original), Foster in view of Davis further discloses the limitation wherein said X-package includes a substantially uniform set of attributes that allows said at least

Art Unit: 2192

one software package to be managed in a single user interface (see, for example, column 7, lines 35-45, which shows the control file associated with a package having a uniform set of attributes, and lines 55-63, which shows a list of such attributes).

With respect to claim 5 (original), Foster in view of Davis further discloses the limitation wherein said package importer tags said X-package with a signature (see, for example, column 11, lines 61-64, which shows a digital signature associated with a package).

With respect to claim 6 (original), Foster in view of Davis further discloses an authentication element to provide verification of the X-package by validating the signature in the X-package with a list of certificates trusted by the target computer (see, for example, column 12, lines 1-5, which shows verifying the authenticity of the package by checking the signature).

With respect to claim 7 (original), Foster in view of Davis further discloses a script extractor to extract an X-package script (see, for example, column 7, lines 14-17, which shows decompressing or extracting the contents of a package, and lines 35-45, which shows the control file that provides a script associated with the package).

With respect to claim 8 (original), although Foster discloses vendor-supplied software packaging systems (see, for example, column 2, lines 8-13), Foster does not expressly disclose the limitation wherein said X-package script includes logic for interacting with a vendor-specific package agent.

However, Davis discloses interacting with vendor-specific packages (see, for example, column 3, lines 44-51, and column 4, lines 1-9), in a system that enables users to install and

Art Unit: 2192

configure third-party applications using a single, common interface (see, for example, column 1, lines 43-57).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the vendor-specific features taught by Davis with the system of Foster, in order to provide a common interface for managing software packages from a plurality of third parties or vendors.

With respect to claim 9 (original), Foster in view of Davis further discloses the limitation wherein said X-package script includes logic for interacting with multiple operating systems (see, for example, column 12, lines 56-67, which shows having platform-independent logic for use with multiple operating systems).

With respect to claim 10 (original), Foster in view of Davis further discloses the limitation wherein said X-package further includes a name of a user who imported said at least one software package (see, for example, column 8, lines 31-33, which shows including the name of a user who created the package, i.e. an importing user).

With respect to claim 11 (original), Foster in view of Davis further discloses the limitation wherein said X-package further includes a hash of package files included in said at least one software package (see, for example, column 11, line 61 to column 12, line 10, which shows using an encryption mechanism to protect against and identify any tampering of the files in a package, i.e. using a hash).

With respect to claim 12 (original), Foster in view of Davis further discloses the limitation wherein said package agent checks relevant operating system of said at least one software package (see, for example, column 10, lines 53-61, which shows checking the operating system to determine whether the software package is compatible).

With respect to claim 13 (original), Foster in view of Davis further discloses the limitation wherein said package agent downloads any needed files (see, for example, column 12, lines 44-46, which shows downloading any needed files).

With respect to claim 14 (original), Foster in view of Davis further discloses the limitation wherein said package agent reports status (see, for example, column 10, lines 57-61, which shows reporting the status of the package).

With respect to claim 15 (previously presented), Foster discloses a software management system (see, for example, the abstract), comprising:

(a) a distribution management server (see, for example, column 12, lines 13-24, which shows a remote distribution source or server); and

(b) a plurality of target computers (see, for example, column 12, lines 13-14, which shows a local client computer, i.e. a target computer, and FIG. 1, which shows that the target computer exists in a network environment, such as one of a plurality of target computers), each target computer including:

(i) an authentication element to provide verification of the X-package by validating the signature in the X-package with a list of certificates trusted by the target computer (see,

for example, column 12, lines 1-5, which shows verifying the authenticity of the package by checking the signature);

(ii) a script extractor to extract an X-package script (see, for example, column 7, lines 14-17, which shows decompressing or extracting the contents of a package, and lines 35-45, which shows the control file that provides a script associated with the package);

(iii) a package agent to receive, deploy and execute said X-package at the target computer (see, for example, FIG. 4 and column 9, lines 19-46, which shows the installation, i.e. the deployment and execution, of a software package, and column 12, lines 25-35, which shows that the software package may be received from a remote server).

Although Foster discloses creating and distributing a software package (see, for example, column 3, lines 47-52), as well as the means to install, upgrade and remove a package based on a control file (see, for example, column 7, lines 46-51) that provides a script comprising a list of commands (see, for example, column 8, lines 39-49), Foster does not expressly disclose a vendor package template that provides a script to install, upgrade and remove the package, and a package importer to create an X-package document based on the vendor package template in preparation for the distribution and installation of the software.

However, Davis discloses a third-party component configuration file, i.e. a vendor package template (see, for example, column 3, lines 44-51), and the means to import the configuration file or template to create a corresponding list of installation tasks, i.e. a script, in preparation for later installation (see, for example, column 4, lines 1-9 and 20-24, and column 5, lines 28-32). The packages are manageable in a system that enables users to install and configure third-party applications using a single, common format or interface (see, for example,

Art Unit: 2192

column 1, lines 43-57), independent of the installation process that is specific to the third party or vendor (see, for example, column 1, lines 22-34).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the vendor package template and package importing features taught by Davis with the system of Foster, in order to provide a common interface for managing software packages from a plurality of third parties or vendors.

Foster in view of Davis further discloses the limitations wherein said script comprises a list of commands in a programming language (see, for example, column 12, lines 56-67, which shows using a computer programming language such as Perl so that the same script can execute on many platforms without recompilation), and wherein said package importer tags said X-package with a signature (see, for example, column 11, lines 61-64, which shows a digital signature associated with a package).

With respect to claim 16 (original), although Foster discloses vendor-supplied software packaging systems (see, for example, column 2, lines 8-13), Foster does not expressly disclose the limitation wherein said at least one software package includes packages from different vendors.

However, Davis discloses an installation system that includes packages from a plurality of third parties or vendors and enables users to manage them using a single, common interface (see, for example, column 1, lines 43-57).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to include packages from different vendors in the Foster system, in order to provide a

common interface for managing software packages from the plurality of sources as taught by Davis.

With respect to claim 17 (original), Foster in view of Davis further discloses the limitation wherein said X-package includes a substantially uniform set of attributes that allows said at least one software package to be managed in a single user interface (see, for example, column 7, lines 35-45, which shows the control file associated with a package having a uniform set of attributes, and lines 55-63, which shows a list of such attributes).

With respect to claim 18 (previously presented), Foster discloses a method for distributing vendor-specific software to target computers (see, for example, the abstract), comprising:

- (a) transferring an X-package to target computers (see, for example, column 12, lines 25-35, which shows transferring a software package from a remote source to a target computer); and
- (b) processing an X-package script (see, for example, column 9, lines 19-46, which shows processing a control file that provides a script).

Although Foster discloses creating and distributing a software package (see, for example, column 3, lines 47-52) having a control file that provides a script comprising a list of commands (see, for example, column 7, lines 35-45, and column 8, lines 39-49), Foster does not expressly disclose importing the vendor-specific software using a vendor package template to create an X-package having a script in preparation for the distribution and installation of the software.

However, Davis discloses a third-party component configuration file, i.e. a vendor package template (see, for example, column 3, lines 44-51), and importing the configuration file

Art Unit: 2192

or template to create a corresponding list of installation tasks, i.e. a script, in preparation for later installation (see, for example, column 4, lines 1-9 and 20-24, and column 5, lines 28-32). The packages are manageable in a system that enables users to install and configure third-party applications using a single, common format or interface (see, for example, column 1, lines 43-57), independent of the installation process that is specific to the third party or vendor (see, for example, column 1, lines 22-34).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the vendor package template and package importing features taught by Davis with the system of Foster, in order to provide a common interface for managing software packages from a plurality of third parties or vendors.

Foster in view of Davis further discloses the limitation wherein said script comprises a list of commands in a programming language (see, for example, column 12, lines 56-67, which shows using a computer programming language such as Perl so that the same script can execute on many platforms without recompilation).

With respect to claim 19 (original), Foster in view of Davis further discloses authenticating the X-package by validating a signature on said X-package (see, for example, column 12, lines 1-5, which shows verifying the authenticity of the package by checking the signature).

With respect to claim 20 (original), Foster in view of Davis further discloses extracting the script from said X-package for processing (see, for example, column 7, lines 14-17, which

shows decompressing or extracting the contents of a package, and lines 35-45, which shows the control file that provides a script associated with the package).

With respect to claim 21 (original), Foster in view of Davis further discloses the limitation wherein said processing said X-package script includes checking a relevant operating system of the vendor-specific software (see, for example, column 10, lines 53-61, which shows checking the operating system to determine whether the software package is compatible).

With respect to claim 22 (original), Foster in view of Davis further discloses the limitation wherein said processing said X-package script includes downloading all relevant files (see, for example, column 12, lines 44-46, which shows downloading all relevant files).

With respect to claim 23 (original), Foster in view of Davis further discloses the limitation wherein said processing said X-package script includes reporting status (see, for example, column 10, lines 57-61, which shows reporting the status of the package).

With respect to claim 24 (previously presented), Foster discloses an apparatus comprising a machine-readable storage medium having executable instructions (see, for example, mass storage 112 in FIG. 1, and column 12, lines 56-67, which shows executable instructions) operable to cause one or more machines to perform operations comprising:

- (b) transfer an X-package to target computers (see, for example, column 12, lines 25-35, which shows transferring a software package from a remote source to a target computer); and
- (c) process an X-package script (see, for example, column 9, lines 19-46, which shows processing a control file that provides a script).

Art Unit: 2192

Although Foster discloses creating and distributing a software package (see, for example, column 3, lines 47-52) having a control file that provides a script comprising a list of commands (see, for example, column 7, lines 35-45, and column 8, lines 39-49), Foster does not expressly disclose operations comprising:

(a) import vendor-specific software using a vendor package template to create an X-package having a script in preparation for later software distribution and installation, said import comprising creating said X-package in a format that makes said X-package manageable in a software package management system independent of vendor-specific aspects of the at least one software package.

However, Davis discloses a third-party component configuration file, i.e. a vendor package template (see, for example, column 3, lines 44-51), and importing the configuration file or template to create a corresponding list of installation tasks, i.e. a script, in preparation for later installation (see, for example, column 4, lines 1-9 and 20-24, and column 5, lines 28-32). The packages are manageable in a system that enables users to install and configure third-party applications using a single, common format or interface (see, for example, column 1, lines 43-57), independent of the installation process that is specific to the third party or vendor (see, for example, column 1, lines 22-34).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the vendor package template and package importing features taught by Davis with the system of Foster, in order to provide a common interface for managing software packages from a plurality of third parties or vendors.

Art Unit: 2192

Foster in view of Davis further discloses the limitation wherein said script comprises a list of commands in a programming language (see, for example, column 12, lines 56-67, which shows using a computer programming language such as Perl so that the same script can execute on many platforms without recompilation).

With respect to claim 25 (previously presented), Foster in view of Davis further discloses the limitation wherein the operations further comprise:

(a) authenticate the X-package by validating a signature on said X-package (see, for example, column 12, lines 1-5, which shows verifying the authenticity of the package by checking the signature).

With respect to claim 26 (previously presented), Foster in view of Davis further discloses the limitation wherein the operations further comprise:

(a) extract the script from said X-package for processing (see, for example, column 7, lines 14-17, which shows decompressing or extracting the contents of a package, and lines 35-45, which shows the control file that provides a script associated with the package).

5. Claim 27 is rejected under 35 U.S.C. 103(a) as being unpatentable over Foster in view of Davis, as applied to claim 1 above, and further in view of U.S. Pat. No. 6,381,742 to Forbes et al. (art of record, "Forbes").

With respect to claim 27 (previously presented), Foster in view of Davis does not expressly disclose the limitation wherein said X-package document includes Extensible Markup Language (XML).

However, Forbes discloses software package manifests or documents implemented using Extensible Markup Language according to an open specification for describing software (see, for example, column 12, lines 29-48).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to include Extensible Markup Language, such as taught by Forbes, in the packages disclosed by Foster in view of Davis, for the purpose of providing support for an open standard.

6. Claim 28 is rejected under 35 U.S.C. 103(a) as being unpatentable over Foster in view of Davis, as applied to claim 2 above, and further in view of U.S. Pat. No. 5,950,010 to Hesse et al. (art of record, "Hesse").

With respect to claim 28 (previously presented), although Foster discloses protecting packages from unauthorized access (see, for example, column 11, lines 64-67), Foster in view of Davis does not expressly disclose the limitation wherein said package importer verifies the importing user identity by checking an access control list.

However, Hesse discloses verifying a user identity by checking a list of users, i.e. an access control list, and determining the security level of the user (see, for example, column 12, lines 19-29), in a system for building and installing custom application packages (see, for example, the abstract).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to protect the packages from unauthorized access, in the software management system of Foster in view of Davis, by checking an access control list, such as taught by Hesse, to verify the importing user identity.

*Conclusion*

7. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael J. Yigdall whose telephone number is (571) 272-3707. The examiner can normally be reached on Monday through Friday from 7:30am to 4:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Art Unit: 2192

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

MY

Michael J. Yigdall  
Examiner  
Art Unit 2192

mjy

  
TUAN DAM  
SUPERVISORY PATENT EXAMINER